Feature and Pose Constrained Visual Aided Inertial Navigation for Computationally Constrained Aerial Vehicles

Brian Williams, Nicolas Hudson, Brent Tweddle, Roland Brockers, Larry Matthies

***Abstract*— A Feature and Pose Constrained Extended Kalman Filter (FPC-EKF) is developed for highly dynamic computationally constrained micro aerial vehicles. Vehicle localization is achieved using only a low performance inertial measurement unit and a single camera. The FPC-EKF framework augments the vehicle's state with both previous vehicle poses and critical environmental features, including vertical edges. This filter framework efficiently incorporates measurements from hundreds of opportunistic visual features to constrain the motion estimate, while allowing navigating and sustained tracking with respect to a few persistent features. In addition, vertical features in the environment are opportunistically used to provide global attitude references. Accurate pose estimation is demonstrated on a sequence including fast traversing, where visual features enter and exit the field-of-view quickly, as well as hover and ingress maneuvers where drift free navigation is achieved with respect to the environment.**

I. INTRODUCTION

To enable the autonomous operation of micro air vehicles (MAV) (Figure 1), onboard estimation of the vehicle position and attitude is required. However, the small payload budgets and physical dimensions of these vehicles severely limit both the sensors which can be used and the computational power available for onboard estimation. Typically, mass constrained MAV-size-vehicles can only use low-grade inertial sensors, which can only be integrated for a few seconds before state estimates significantly diverge. By augmenting the estimate with observations from a light-weight camera though this inertial-only estimate can be greatly improved [1], [2], [3], [4]. A good introduction to the complementary nature of inertial and visual sensors is provided in [5]. In this paper we present a navigation system for a MAV which incorporates measurements from both inertial and visual sensors with particular attention to the low onboard computational budget.

The navigation system required must be able to perform in the following three flight scenarios:

- Fast Traverse – When flying between locations of interest, the MAV will move through unexplored terrain and new visual features will pass out of the field-of-view of the camera soon after they are first seen. The system should be able to track the general motion of the vehicle in this scenario but does not need to maintain a representation of the environment that passes out of view.

Fig. 1. Asctec Pelican Quadrotor approaching a building for ingress.

- Hovering – When hovering, the same visual features will stay in view for longer periods of time. The position estimate of the system should not drift in this scenario.
- Building Ingress – The MAV must be able to navigate relative to a door or window sufficiently accurately to fly through the opening and into the building. Due to the limited field-of-view of the camera, the edges of the opening may not be in view during the entire ingress.

Pose estimation must be computationally cheap enough for a real-time implementation onboard the MAV, and the estimate must be sufficiently accurate to allow reliable operation.

Many real-time pose estimation algorithms have been developed which could be applicable to some of the scenarios described above. Perhaps the best real-time monocular camera tracking system developed is the Parallel Tracking and Mapping (PTAM) system developed by Klein and Murray [6]. This system performs bundle adjustment over keyframes to build a map of point landmarks in the environment. In a parallel processing thread, the camera pose is tracked relative to this map. The system performs very well but is limited to small environments as the computation time becomes prohibitive as the number of keyframes grows. Even so, this system has been demonstrated for navigation of a MAV [1] but due to the high computation cost of bundle adjustment, the estimation was performed on a ground station rather than onboard.

By comparison, filtering is more suited to real-time systems with low computation power [7]. Filtering also makes the incorporation of data from an inertial measurement unit (IMU) straightforward which allows fast position updates

B. Williams, N. Hudson, R. Brockers and L. Matthies are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA {bpwillia,nhudson,brockers,lhm}@jpl.nasa.gov

B. Tweddle is with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA tweddle@mit.edu

(100Hz) and makes the true scale of the camera motion observable.

In general, there are two filtering approaches for vision based navigation: pose-based and feature-based. These filters augment the system state with either previous camera poses or the location of observed visual features, respectively. As computational complexity limits the size of the filter state, choosing the type of data to include in the filter state leads to different filter behaviors.

Pose-based filtering [8], [9], [10], referred to here as Vision Aided Inertial Navigation (VAIN), is in many respects similar to bundle adjustment. VAIN systems augment the filter state with the current estimate of the camera position when an image is taken, and builds up a set of previous camera poses over time. The filter complexity is only linear in the number of observed features, which are tracked between previous camera poses, and then used to update the vehicle state. This allows the inclusion of a very large number of observed features, which in turn can result in very accurate motion estimation. For this reason, we have chosen to use the system of Mourikis and Roumeliotis [10] as the basis for our own system. When the size of the state becomes too long, the older poses are removed from the state, which allows the system to estimate the motion of the vehicle even as it continually explores. However, once a pose is removed from the state, and the measurements of features associated with that camera pose are used, no further reference back to the observation can be made. This can lead to filter drift, even with sustained observations of the same feature set.

Feature-based filtering [11], [12], [13], [14] instead estimates the position of visual features in the world and estimates the current vehicle pose relative to this map of features. As long as these features stay in the state vector and are observed, the pose estimate relative to these features will not drift. However, for computational reasons these systems are limited in the number of features they can maintain in the state. If sparse feature maps are used, fewer observations are used in each frame and so the pose estimate of the vehicle has fewer constraints and is therefore less accurate. Furthermore, in a monocular system, the depth of a newly observed feature is not immediately observable and so the full feature state cannot be estimated immediately. Either the features must be initialized in a separate filter [14], [15] in which case some observations of the feature are not used to constrain the vehicle pose, or the feature is placed immediately in the state vector with a prior on the depth [16], which can bias the estimate. For instance, the work by Jones et. al. [14] initializes each new feature in a sub-filter until the feature covariance drops below a specified bound, and the feature is added to the main filter state. Only after initialization does the visual information affect the vehicle state. This delay can lead to poor performance in scenarios where features might quickly pass trough the field-of-view.

Instead, the navigation system proposed in this paper is a hybrid between a pose-based and a feature-based filtering system. Like other systems, the IMU measurements are integrated to produce an up-to-date pose estimate at the high IMU measurement rate. This integrated pose also acts as the prior estimate for the vehicle pose when the next camera frame is recorded. A camera pose is added to the filter state when a new frame is recorded, and the current camera view has sufficient baseline to the previous pose. A large number of visual observations between the poses in the state vector are used to constrain the pose estimates very accurately. Like other pose-based systems, these visual features are forgotten as soon as their measurements have been used to constrain the poses in the sliding window. We call these *Opportunistic Features* (OFs). In addition to this pose-based estimation, our system includes a few select visual features, called *Persistent Features* (PFs) into the filter state. Persistent features are selected once it is ascertained that the feature can be tracked, with preference given to features that are likely to remain in the field-of-view. As such, the PF can be easily triangulated from the previous camera frames, and then added into the system state without significant bias. Two types of feature are added to the state in this way: visual corner features parameterized as a Cartesian point in 3D, and vertical line features denoting the left and right edge of a door or window which the MAV intends to fly through. These persistent features perform two functions. Firstly, they allow the vehicle to estimate its position relative to an opening as it performs an ingress. Secondly, when the vehicle is hovering and keeps the same features in view the vehicle pose estimate will not drift. The number of persistent features is kept very low to decrease the computational burden. Typically the position of a few Cartesian PF are tracked (max five), along with edges of the opening for the next ingress. This feature-and-pose constrained filter is able to achieve the benefits of both approaches: accurate motion estimation during traverses with a large number of observations acting as constraints, and drift free behavior when hovering or maneuvering through an opening in a building.

In urban situations, the FPC-EKF also makes use of opportunistic observations of vertical edges, to provide a global attitude reference. A direct EKF updated based on vertical edges measurements was developed, which does not require any state augmentation.

II. Estimation Framework

The Feature-and-Pose Constrained Extended Kalman Filter (FPC-EKF) described here estimates the pose of a 6 degree of freedom micro air vehicle as it flies through the environment, hovers, and passes through doors and windows. The estimation framework incorporates measurements from both an inertial sensor and a camera. The system is particularly suited to real-time onboard operation for an aerial vehicle performing ingress maneuvers as it incorporates ideas from both pose-based and feature-based techniques in the same filtering framework. While the pose-based tracking framework allows incorporation of hundreds of visual feature observations, feature-based aspects of the estimator then allow drift free tracking while hovering and the ability to maneuver relative to an opening. In addition, observations of vertical edges which are common in man made environments, are used to improve the attitude estimate of the vehicle in flight.

The vehicle state is parameterized by a quaternion representation, q_{bg}, of the ground frame G rotation with respect to the vehicle body frame B, the vehicle position p_{gb} and velocity v_{gb} with respect to G, as well as the accelerometer b_a and gyroscope b_ω biases:

$$X = \begin{bmatrix} q_{bg}^T \; b_\omega^T \; v_{gb}^T \; b_a^T \; p_{gb}^T \end{bmatrix} \quad . \tag{1}$$

For mathematical convenience, the vehicle body frame is defined to be coincident with the IMU frame. While the full vehicle state utilizes a four-component quaternion representation of attitude, the covariance of the system is represented as a three-component representation of attitude errors [17]. This is the same state representation approach as used by Mourikis and Roumeliotis [10].

The FPC-EKF augments the vehicle state (1) with a selection of previous camera poses, a few select point landmarks to reduce drift, and an estimate of ingress points when one is identified. Each aspect of pose estimation system will be discussed in turn in the following sections.

A. Inertial Measurements

The vehicle carries an inertial measurement unit (IMU) providing both measurements from both three-axis accelerometers and gyroscopes. The measured acceleration and angular velocity at each time-step are used to update the current estimate of the pose and velocity of the vehicle in a standard integration process. The vehicle state dynamics [10] described below are integrated analytically over the IMU sample period to provide the discrete time EKF state propagation:

$$\begin{aligned} \dot{q}_{bg} &= \frac{1}{2}\Omega(\omega)q_{bg}, \quad \dot{v}_{gb} = a_g \\ \dot{p}_{gb} &= v_{gb}, \quad \dot{b}_a = n_a \quad \dot{b}_\omega = n_\omega \end{aligned} \tag{2}$$

where a_g is the vehicle acceleration in the ground frame, ω is the rotational velocity in the body frame, and n_a and n_ω are zero mean Gaussian noise processes associated with the IMU bias drift, and:

$$\Omega(\omega) = \begin{bmatrix} -\omega\times & \omega \\ \omega^T & 0 \end{bmatrix}, \quad \omega\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad . \tag{3}$$

The IMU gyroscope and acceleration measurements ω_m and a_m respectively are used along with the current state estimates to approximate the ground frame acceleration a_g and body frame rotation rates ω:

$$\hat{a}_g = R(q_{bg})^T (a_m - b_a) \tag{4}$$

$$\hat{\omega} = \omega_m - b_\omega \tag{5}$$

Note that the angular rate measurement does not compensate for the rotation of the earth, as is typically done with tactical grade IMU navigation. This simplification was made for two reasons, first the IMU sensor is very low grade, meaning accelerations due to the Earth's rotation are insignificant compared to integration times, bias drift and sensor noise. Secondly, due to operational constraints, the ground frame's location and orientation with respect to the earth may be unknown.

For computation reasons, the estimates $\hat{a}_g$ and $\hat{\omega}$ in Equations (4)-(5) are assumed to be constant over the IMU sample period (1/100 s). Unlike the approach of Mourikis and Roumeliotis [10], this allows the analytic integration of the dynamics equation (2) using standard linear matrix differential equations, and analytic integration of the associated covariance propagation equation. Comparison with a numerical integration approach showed no resulting effect on estimator performance.

B. Pose-Based Visual Constraints

The filter incorporates previous camera poses into the system state to provide an efficient way of utilizing hundreds of observed opportunistic features (OFs) to constrain the vehicle's motion estimate. The utilized pose-based framework is substantially similar to the Multi-State Constraint Kalman Filter (MSCKF) developed by Mourikis and Roumeliotis [10], and readers should refer to their work for implementation details. A brief description will be given here for convenience and to explain how it fits into the rest of our system. The MSCKF framework was chosen as the computation complexity is only linear with respect to the number of OF used in the filter, and it allows incorporation of OF tracked over multiple camera frames as opposed to just two-frame comparisons [8], [9]. However, this filter has complexity at least quadratic [10] in the number of frames added to the system state.

When a new frame is collected by the camera, the estimated pose (attitude and position) of the camera at that moment is cloned from the vehicle state and used to augment the state vector, a process termed "stochastic cloning". To meet computational constraints, the maximum number of frames in the system state is limited to M_{max}. Once M_{max} is reached, older frames in the state vector must be removed and are no longer used for updating.

In each frame, observations are made of up to several hundred opportunistic feature points, using a sub-pixel precise version of the STAR feature detector, a center-surround type detector, for detecting blob like features, based on [18] and [19], and correspondences between frames are found using upright SURF [20].

To maintain filter consistency, each measurement of an OF is used only once in the MSCKF filter. Opportunistic features measurements are not necessarily used in the Kalman filter update if they are observed in the current frame. Instead OFs are only included in the filter update according to either of two criteria: First, when the feature leaves the field-of-view, or is no longer tracked, and second if the maximum number of frames M_{max} is reached, all the features associated with any removed frame are used in the filter update.

To use a OF in the filter update, the feature position in the global frame f_g is first triangulated using Levenberg-Marquardt (LM) minimization, where the reprojection error between f_g and the camera poses in the state are minimized. The EKF measurement equation is then created by linearizing the predicted feature projection in each camera about f_g and the previous state camera poses.

To avoid inconsistencies from including f_g into the measurement equation, the linearized measurement equations are reduced by projecting the measurements onto the left nullspace of the Jacobian for the feature location.

Unlike [10], we do not keep a pose in the state vector for every new camera frame. Instead, our system makes more efficient use of limited computational power and adds a new frame only if it has a significant baseline from the previous frame stored in the state. This is done by performing a stereo registration of the common feature observations between the two views using the rotation estimate and looking for significant disparity.

C. Feature-Based Visual Constraints

The system keeps a small number of *Persistent Features* (PFs) in the EKF state vector at all times. If the vehicle then slows to a hover, the vehicle position estimate will not drift relative to these features as they continue to be observed. Like OFs, PFs are feature points observed by the camera, which are picked randomly from observed features. The estimate of the feature position in the world is parameterized as simply 3D cartesian point. New PFs are added to the EKF state fully initialized rather than going through an initialization phase like [15] or by using a prior on depth like [16]. Our system takes advantage of the vehicle poses and their associated images already estimated in the state. When the number of PFs currently visible drops below a threshold (we have found five to be sufficient), a new PF is selected from the potential OF measurements which have not yet been used. The system chooses a new feature along the direction of motion of the vehicle and with sufficient baseline to allow a good initialization. The initial estimated position for the new PF is obtained by triangulating its location from two camera views already present in the filter state for the pose-based estimation. The PF is placed in the state with correlations due to the poses used to calculate its initial position. If the feature is also visible in the other frames in the state then these observations are used to update the EKF state. In all later timesteps, observations of the PFs are used to constrain the current vehicle pose in the standard way for feature-based filters. When any of the PFs passes out of view of the camera it is removed from the state to make room for new PFs. A persistent map is not necessary for our application.

D. Ingress Features

It is important for the system to be able to accurately navigate relative to a door or window when the MAV is flying into a building. For this reason, the position of the ingress target must be estimated in the same filter.

To identify a building entrance, the system detects rectangular wall openings, such as windows or doors, as a gap inside a wall surface bounded by straight lines. In our experiment we concentrate on detecting door openings as ingress targets, which allows us to simplify the tracking problem. As it is often difficult to accurately track the upper and lower horizontal boundary of the door opening during approach, because they often are not in view when the vehicle is centered in front of the opening, we concentrate on tracking the vertical edge positions as the most crucial component for a door ingress maneuver and neglect accurate height tracking of the ingress target. To initially detect a door opening, a homography based surface reconstruction (cp. [21]) is used to separate in-plane features that are located on the planar wall surface from off-plane features that are assumed to be located behind it (Figure 2). A histogram based clustering of image regions with in-plane and off-plane features extracts candidate regions for wall openings. If these regions are bounded by vertical straight lines, which are detected using Canny edge detection and a Hough transform, a door opening is detected and the position of the left and right edge is added to the filter state as soon as possible.

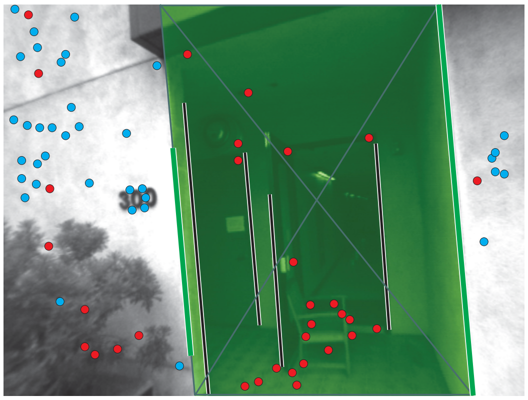


Fig. 2. Door detection for ingress: an area of out-of-plane features (red dots) bounded by vertical line segments (green). In-plane features are labeled blue, other detected vertical lines are labeled black.

When two stored poses in the state observe the door with sufficient baseline, the two vertical edges on either side of the ingress target are triangulated. Each vertical line is parameterized as an infinite line passing through a point on the global horizontal plane. The 2D coordinates of this point are put into the EKF state. This triangulation process is illustrated in Figure 3.

Once the ingress target has been added to the state, further observations of the two edges can be used to improve the estimate. In all new frames where the ingress target is visible, the vertical edges are tracked and used for the EKF update. The measurement used by the EKF is the perpendicular distance of the two observed endpoints of each line from the predicted projection of the infinite line into the image. Data association for the two vertical edges being estimated is done using intensity histograms for each side of the line.

If the estimated position of the ingress point is not maintained by the filter along with the vehicle pose then the estimate of their relative position can drift over time. This would make flying through the confined space of an ingress

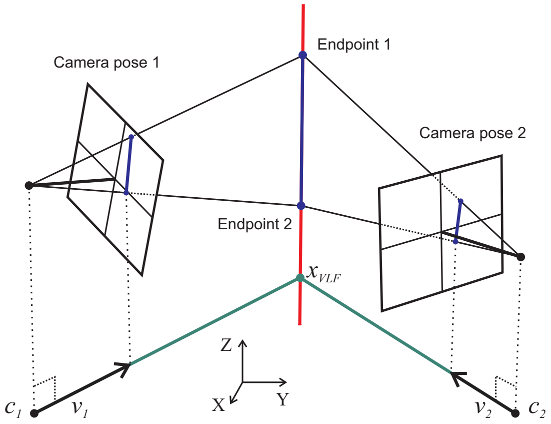


Fig. 3. During ingress, the vertical edges on either side of the opening are triangulated and subsequently estimated by the EKF. The edges are parameterized as a point x_{VLF} on the global horizontal plane (X-Y). This point is determined from the crossing point of the projection onto this plane of the rays (v_1 and v_2) defined by the observation of the lines in each camera (c_1 and c_2).

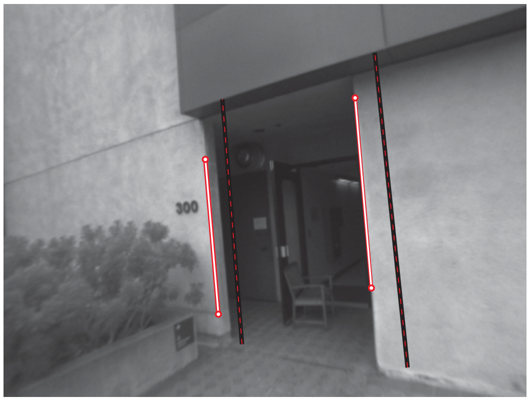


Fig. 4. If the position of the ingress point is not estimated in the filter along with the vehicle pose then the estimate of their relative pose can drift. This figure shows the predicted position of the doorway edges (dashed lines) after 30 seconds of drift. By estimating the position of the ingress point in the filter, our system correctly predicts the doorway at the observed position (solid lines).

point dangerous. This drift is illustrated in Figure 4 where the estimated position of the door is not maintained after its initial triangulation. After 30 seconds, the position of the doorway relative to the vehicle has significantly diverged. If the ingress point position estimate is maintained in the filter then the estimated relative position is correct.

E. Vertical Line Measurements

Apart from the lines on either side of the ingress point, many more vertical lines are often visible in man made environments. Though the exact location of these lines is not needed for navigation, measuring vertical lines in the image can improve the filter's estimate of the vehicle's roll and pitch.

At each frame, the vertical lines are detected using Canny edge detection and the Hough transform. Those not corresponding to the current ingress target are used for these measurements. The depths of these lines are never calculated and their correspondence between frames is not used. Non-vertical lines are excluded using a Mahalanobis compatibility test.

A set of parallel lines in the world point towards the same vanishing point when projected into an image. The perpendicular distance of the observed vertical lines and the vanishing point predicted by the current estimated vehicle orientation is used as the measurement for the EKF update. If the vehicle orientation estimate is correct then the observed vertical lines will point directly towards the predicted vanishing point and the measured perpendicular distance would be zero.

The calculation of a vanishing point is best performed in homogeneous coordinates. Let $\mathbf{R}_{CG}$ represent the a priori estimates of the rotation matrix from the global to the camera frame. The camera projection matrix is denoted as $\mathbf{\Pi}$, which is the three-by-four element matrix that maps from three dimensional homogeneous coordinates to two dimensional homogeneous coordinates. The homogeneous coordinates for the point at infinity corresponding to vertical (z-axis aligned) lines is $\mathbf{z} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T$. Using this the coordinates for the vanishing point $\mathbf{v}$ can be computed as follows:

$$\mathbf{v} = \mathbf{\Pi}\mathbf{R}_{CG} \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0} \end{bmatrix} \mathbf{z} \tag{6}$$

Note that the position of the vanishing point does not depend on the translation of the camera since we are projecting a point at infinity.

The distance, d, between the predicted vanishing point location, $\hat{\mathbf{v}}$, and each observed vertical line, $\mathbf{l}$ is the dot product between the two expressed in normalized homogeneous coordinates.

$$d = \mathbf{l}_{normalized} \cdot \hat{\mathbf{v}}_{normalized} \tag{7}$$

The difference between this measured distance and the predicted value of 0 is used as the innovation in the EKF update step to improve the estimate of the vehicle's roll and pitch relative to gravity. Without these measurements, the vehicle's attitude is still observable due to the gravity component of the acceleration measured by the IMU. However, this estimate alone can be quite noisy and is greatly improved when these visual observations are also taken into account. This is illustrated in Figure 5 where the estimated horizon line is drawn in the image with (green solid) and without (red dashed) the observations of the vertical lines (black) being used by the filter.

III. MONTE CARLO OPTIMIZATION OF FILTER

The ultimate goal in combining pose and feature-based constraints into the filter is to benefit from the advantages of both approaches. Pose-based filtering allows information from many features to be incorporated efficiently, and can utilize only short feature tracks, while feature-based filtering provides drift free localization when at least three features can be persistently tracked. In theory, one would expect the

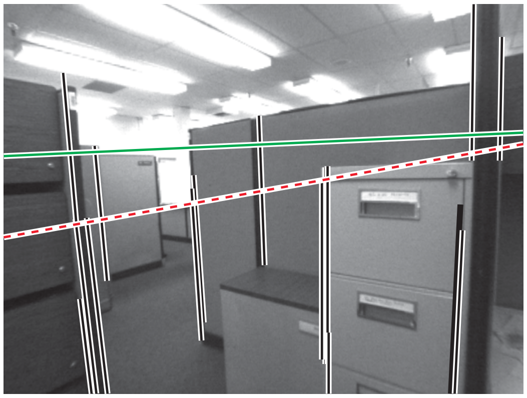


Fig. 5. Vertical line observations improve the estimate of the vehicle's roll and pitch. Here, the estimated horizon line is drawn on the image with (green solid line) and without (red dashed line) observations of vertical lines (black solid) being used.

performance of the filter to increase with both the maximum number of poses added to the system state M_{max}, and the number of features included in the state vector N. At the same time, the maximum state vector length is then equal to $15+3N+6M_{max}$, which naively implies that computational complexity is $O((N+M_{max})^3)$.

Monte Carlo (MC) simulations were used to investigate the effect of changing N and M_{max}, in terms of computational cost and the root mean square error (RMSE) of the filter estimate compared to the simulated ground truth. 1000 Monte Carlo runs were used to simulated relevant mission profiles, including periods of fast motion and hover conditions. Each simulation created a set of noisy observations (OFs, PFs, and inertial measurements), which assumed optimal tracking (known data association, and features were tracked as long as they were in view). Based on this set of noisy measurements, a set of filters was run, each with different N and M_{max}.

A unique revelation about MSCKF-pose-based filtering is that increasing the number of frames M_{max} did not produce uniformly better results (Figure 6), as is implied in [10]. Instead filter error eventually increased with respect to M_{max}. After investigation, we believe that M_{max} needs to be made consistent with the IMU integration performance. In fact increasing M_{max} in practice tends to make the updates more precise (as they linearize around a larger baseline), but less frequent, as the filter aggregates measurements up to M_{max} (see Section II-B and [10]). The low-performance IMU utilized here can only be integrated for less than a second before pose estimates errors become significant. Unless numerous vision based filter updates are made every second, overall filter performance degrades. This effect is probably not seen in [10] due to the use of a high performance IMU system.

The implication of this results is that small M_{max} is optimal from both a computational and a performance perspective for MAV systems with low performance IMUs. In this case a fixed M_{max} allows the addition of N PF tracks to be performed by the filter, where N can be increased up to the computation limits of the system.

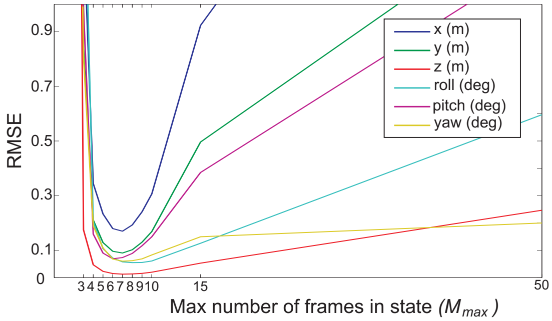


Fig. 6. Monte Carlo simulation of RMSE of vehicle state compared with simulated ground truth for different M_{max}, the maximum number of camera frames allowed in the filter state vector. RMSE did not decrease uniformly with M_{max}, instead an optimal $M_{max} = 7$ was found.

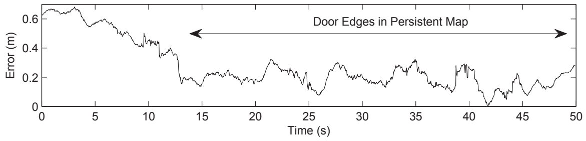


Fig. 7. Camera position error relative to VO in real-world sequence.

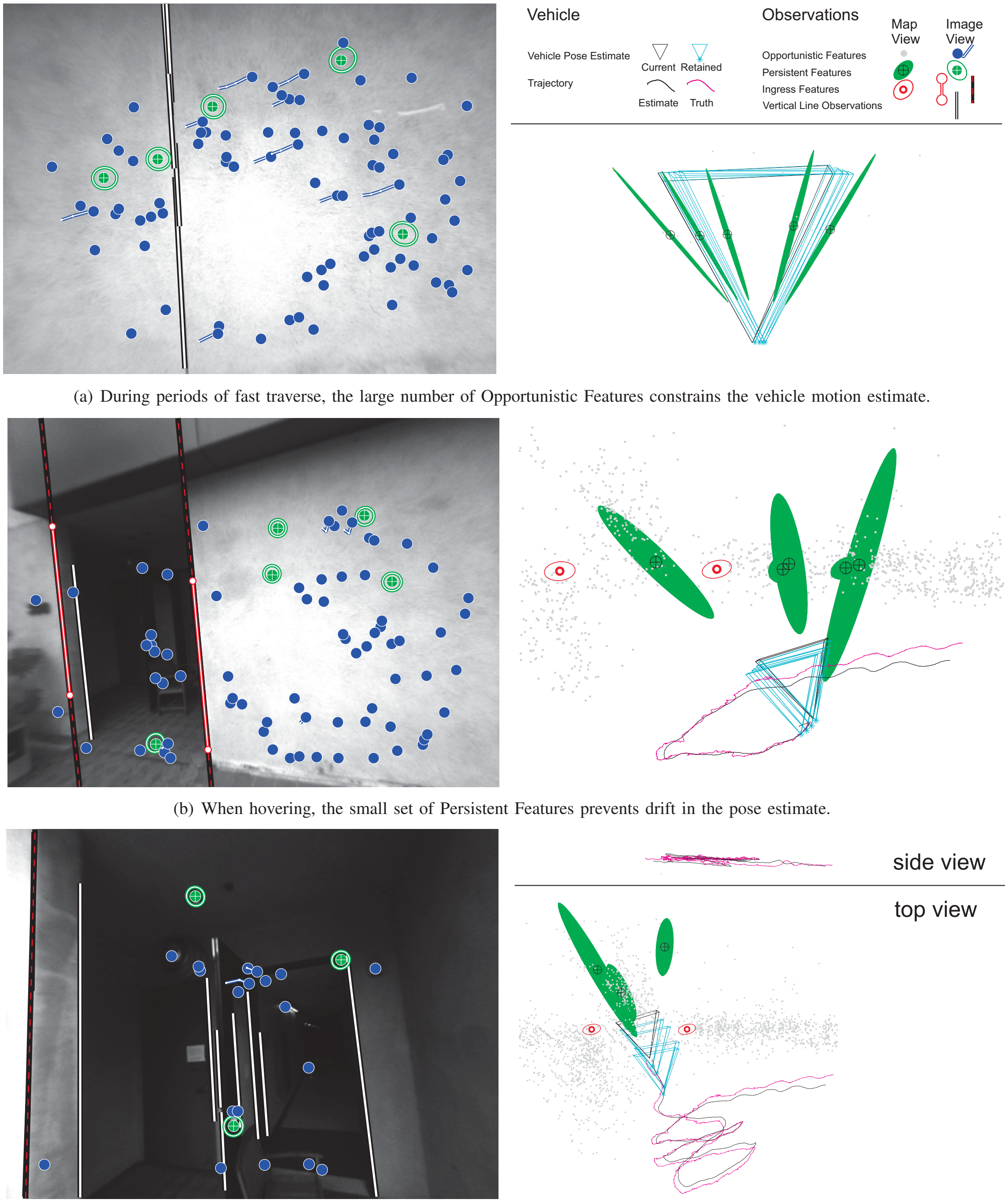
IV. EXPERIMENTAL RESULTS

The performance of the system has been evaluated using a real-world sequence demonstrating all three scenarios of interest: traverse, hovering, and ingress. Results of this experiment are shown in Figure 8. To better assess the performance of the system a Bumblebee stereo rig with attached IMU was used. This allows us to calculate approximate ground truth motion for the sequence using stereo visual odometry (VO). The VO method of Howard [22] was used which has been shown to be accurate to 0.25% of distance covered. Then, to test our algorithm, the images from just one side of the stereo pair were used since the MAV itself supports only a single camera. The IMU used in the experiments was a MicroStrain 3DM-GX1.

At the beginning of the sequence, a traverse was demonstrated as it translated along the wall to the right of the doorway. As is shown in Figure 8(a), the large number of OFs observed during a fast traverse are key to constraining the vehicle motion estimate. During these periods, visual features are only in view for a short period of time and so PFs are less important here. Vertical lines observed during this time help prevent drift in the attitude estimate of the vehicle.

When the door comes into view it is identified as an ingress point (Figure 2) and is added to the filter state. Soon after, the system slows to a stop in front of the doorway and drift free tracking is demonstrated relying on observations of PFs in the map (Figure 8(b)). During a near stationary hover like this, OFs are of little help since there is no baseline in successive frames to allow triangulation.

Figure 8(c) shows the estimate as the system performs the ingress maneuver and passes through the doorway. Since the



(a) During periods of fast traverse, the large number of Opportunistic Features constrains the vehicle motion estimate.

(b) When hovering, the small set of Persistent Features prevents drift in the pose estimate.

(c) As the vehicle performs an ingress into a building, the estimate of the boundary of the ingress point allows the vehicle to avoid colision.

Fig. 8. The system was tested in a real-world scenario demonstrating (a) Fast Traverse, (b) Hovering, and (c) Ingress. The left column shows the camera view at these key moments with corresponding observations. The right column shows the filter estimate. See the key above for symbols used. Uncertainty ellipses of three standard deviations are given for the Persistent and Ingress Features.

position of the ingress point is estimated by the filter, the vehicles position relative to the boundaries are known and the vehicle can safely pass through the doorway.

The ground truth trajectory from VO was aligned to our estimate using a camera pose late in the sequence but before the ingress. Figure 7 shows that the error in vehicle position is bounded while the persistent door features are in the map and in view of the camera. However, during the traverse at the start of the sequence the estimate drifts since PFs are removed from the map as they pass out of view.

The unoptimized system currently requires 140ms for image processing (implemeted in C) and 75ms for state estimation (Matlab) per frame, processed offline on a 3.2GHz desktop machine.

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

The developed Feature and Pose Constrained Extended Kalman Filter (FPC-EKF) allows navigation of micro air vehicles (MAVs) using only an inertial measurement unit and a single camera. The incorporation of both poses and features into the filter state allows the efficient use of hundreds of observed visual features to constrain vehicle motion, while achieving navigation with respect to critical environment features such as ingress points. This hybrid solution avoids the limitations of both pose-based filters, where drift occurs even with sustained viewing of landmarks, and feature-based filters, which use sparse maps and therefore make fewer observations per time-step. The use of Monte Carlo simulation showed the unique applicability to computationally constrained MAVs. Due to low-performance IMU sensors, limiting the maximum number of poses in the filter state was both optimal in terms of filter performance and computation.

The designed filter was run offline on a MAV-relevant data set, where the vehicle traversed along a building before detecting and hovering in front of a doorway, and finally entering into the building. Results show low drift rate tracking during traverse, and sustained localization with respect to critical door edges during hover and ingress.

B. Future Work

Current and future work will focus on implementing the FPC-EKF on-board the Asctec Pelican Quadrotor. Future testing will be conducted to verify both feature detection and tracking in varied terrain, as well as estimator performance. Particular attention will be focused on the trade-off of low cost feature detection and tracking, suitable for pose-based filters, and the limited use of high-performance feature descriptors for sustained tracking of critical features.

VI. ACKNOWLEDGMENTS

The research described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Funding was provided by the Army Research Laboratories, the MAST CTA program, and an appointment for BW to the NASA Postdoctoral Program at JPL, administered by Oak Ridge Associated Universities through a contract with NASA.

REFERENCES

[1] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based MAV navigation in unknown and unstructured environments," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 21–28.

[2] T. Cheviron, T. Hamel, R. Mahony, and G. Baldwin, "Robust nonlinear fusion of inertial and visual data for position, velocity and attitude estimation of UAV," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 2010 –2016.

[3] L. Mejias, P. Campoy, K. Usher, J. Roberts, and P. Corke, "Two seconds to touchdown - vision-based controlled forced landing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006, pp. 3527 –3532.

[4] S. Saripalli, J. Montgomery, and G. Sukhatme, "Vision-based autonomous landing of an unmanned aerial vehicle," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2002, pp. 2799 –2804.

[5] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *International Journal of Robotics Research*, vol. 26, no. 6, 2007.

[6] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007.

[7] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Real-Time Monocular SLAM: Why Filter?" in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2010.

[8] S. Roumeliotis, A. Johnson, and J. Montgomery, "Augmenting inertial navigation with image-based motion estimation," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 4326–4333, 2002.

[9] D. Bayard and P. Brugarolas, "An estimation algorithm for vision-based exploration of small bodies in space," *Proceedings of the 2005, American Control Conference, 2005.*, pp. 4589–4595, 2005.

[10] A. Mourikis and S. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 3565–3572.

[11] A. J. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2007.

[12] P. Piniés, T. Lupton, S. Sukkarieh, and J. Tardós, "Inertial aiding of inverse depth SLAM using a monocular camera," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007, pp. 2797–2802.

[13] J. W. Langelaan, "State Estimation for Autonomous Flight in Cluttered Environments," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1414–1426, 2007.

[14] E. Jones, A. Vedaldi, and S. Soatto, "Inertial structure from motion with autocalibration," in *IEEE 11th International Conference on Computer Vision, Workshop On Dymanical Vision*, 2007.

[15] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. IEEE International Conference on Computer Vision*, 2003.

[16] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[17] F. L. Markley, "Attitude Error Representations for Kalman Filtering," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 2, pp. 311–317, 2003.

[18] M. Agrawal, K. Konolige, and M. R. Blas, "Censure: Center surround extremas for realtime feature detection and matching." in *Proc. Europ. Conf. on Comp. Vis.*, ser. LNCS, vol. 5305, 2008, pp. 102–115.

[19] "Star detector wiki," http://pr.willowgarage.com/wiki/Star_Detector.

[20] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008.

[21] H. Longuet-Higgins, "The reconstruction of a plane surface from two perspective projections," *Proc. Roy. Soc. London Series B 227 (1249)*, pp. 399–410, 1986.

[22] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France (IROS)*, 2008, pp. 3946–3952.